한국 CAD/CAM 학회 논문집
제 11권 제 3호 2006년 6월 pp. 205-210

학술논문

다항식곡선으로부터 순차적 방법에 의한 점열의 생성

주 상 윤*

A Tessellation of a Polynomial Curve by a Sequential Method

Ju, S.Y.*

ABSTRACT

Curve tessellation, which generates a sequence of points from a curve, is very important for curves rendering on a computer screen and for NC machining. For the most case the sequence of discrete points is used rather than a continuous curve. This paper deals with a method of tessellation by calculating the maximal deviation of a curve. The maximal deviation condition is introduced to find the point with the maximal chordal deviation on a curve segment. In the previous research a curve tessellation was tried by the subdivision method, that is, a curve is subdivided until the maximal chordal deviation is less than the given tolerance. On the other hand, a curve tessellation by sequential method is tried in this paper, that is, points are generated successively by using the local property of a curve. The sequential method generates relatively much less points than the subdivision method. Besides, the sequential method can generate a sequence of points from a spatial curve by approximation to a planar curve. The proposed method can be applied for high-accuracy curve tessellation and NC tool-path generation.

Key words : tessellation, maximal deviation condition, subdivision method

1. 서 론

최근 고속 및 고정밀 NC기계에 의하여 제품가공이 이루어지고 있으며 제품의 정밀도에 대한 관심이 매우 높아졌다. 일반적으로 NC가공은 공구가 따라가야 할 곡선으로부터 주어진 허용오차 범위에서 얻어진 점열을 보간한 직선선분들을 따라 이루어진다. CAD/CAM분야에서 주어진 곡선을 점열화하는 방법으로는 크게 2가지 접근법이 이용되고 있다. 첫번째 방법은 곡선 상에서 적절히 점열을 취하여 이웃한 두 점들을 양끝점으로 갖는 구간곡선 상의 중간점으로부터 현까지의 거리를 구하여 그 크기가 허용오차를 벗어나면 중간점을 추가하여 구간을 둘로 분할하는 방법이다[1]. 이러한 방법은 매우 단순하므로 적용하기 쉬운 잇점이 있으나 계산된 최대오차의 크기가 허용오차를 벗어날 수 있다는 문제점을 가진다. 나머지 다른 방법으로는 B-스플라인곡선이나 Bezier 곡선의 convex hull 성질을 이용하는 것으로 현과 곡선 사이의 최대오차를 곡선의 조정점을 이용하여 구하는 방법이다[2]. 즉 현과 곡선의 조정점들까지의 거리를 구하여 최대거리가 허용오차를 벗어나면 곡선에 중간점을 추가하여 곡선을 작게 분할한다. 이 경우 곡선의 조정점을 이용하여 구한 현과의 최대거리는 실제 곡선의 최대오차 값보다 항상 크게 되므로 이 방법을 사용하여 생성된 점열은 허용오차를 항상 만족시키게 된다. 하지만 곡선의 조정점을 이용한 분할방법은 생성되는 점들이 필요 이상으로 많아질 뿐만 아니라 옵셋곡선에서는 적용이 불가능하다는 문제점이 있다.

최근에 최대오차조건식을 이용하여 곡선으로부터 점열을 생성하는 방법이 제안되었다[3]. 최대오차조건을 만족하는 곡선상의 점에서 현과의 최대오차를 구하여 허용오차를 벗어나면 곡선을 분할하는 방법이다. 이 방법은 곡선과 현 사이의 최대오차를 정확하게 구할 수 있을 뿐만 아니라 옵셋곡선에 대해서도 적용할 수 있다는 장점이 있으므로 간섭이 없는 NC데이터를 생성하는데 유용하게 이용될 수 있다. 하지만 이 방법에 대하여 현재까지 연구된 것은 단지 평면다항

*교신저자, 종신회원, 울산대학교 산업공학과
- 논문투고일: 2005. 07. 22
- 심사완료일: 2005. 12. 20

식곡선에 대하여 점열을 생성한 것에 머무르고 있다.

본 논문에서는 최대오차조건식을 이용하여 순차적으로 점열을 생성하는 방법을 제안하고자 한다. 순차적으로 점열을 생성하는 경우 한 점에서 이웃한 다음 점만을 반복적으로 구하게 되므로 이웃한 점들 사이의 짧은 구간곡선을 평면곡선으로 근사시킬 수 있게 되어 공간곡선에 대해서도 근사적으로 최대오차조건식을 적용하여 점열을 얻을 수 있다. 또한 최대오차가 허용오차에 가깝게 되는 점을 구할 수 있게 되어 분할법보다 적은 수의 점을 가진 점열을 기대할 수 있다.

본 논문은 다음과 같이 구성된다. 2장에서는 곡선으로부터 최대오차를 갖는 점을 찾기 위한 최대오차조건식을 유도한다. 3장에서는 먼저 3.1절에서 최대오차조건식을 이용하여 평면곡선으로부터 최대오차를 결정하는 방법을 소개한 후 3.2절에서 공간곡선을 평면곡선으로 근사시켜 최대오차를 결정하는 방법을 제시한다. 4장에서는 이웃한 점들을 연속적으로 찾아가는 순차적 점열생성의 방법을 제시한다. 5장에서는 분할법으로 생성된 점열과 순차법으로 생성된 점열을 비교한 후 마지막 6장에서는 본 논문에 대한 결론을 기술한다.

2. 곡선의 최대오차조건

Fig. 1에서 보는 바와 같이 구간 $[t_i, t_{i+1}]$에서 곡선 $\mathbf{r}(t)$의 오차벡터 $\mathbf{h}(t)$는 다음과 같이 정의된다.

$$\begin{aligned}\mathbf{h}(t) &= \mathbf{r}(t) - \mathbf{Q}(t)\\ &= [\mathbf{r}(t) - \mathbf{P}_i] - \{[\mathbf{r}(t) - \mathbf{P}_i]\cdot\mathbf{u}\}\mathbf{u} \qquad (1)\end{aligned}$$

여기서 $\mathbf{u} = \dfrac{\mathbf{P}_{i+1} - \mathbf{P}_i}{|\mathbf{P}_{i+1} - \mathbf{P}_i|}$

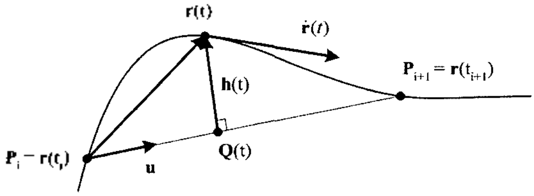


Fig. 1. An error vector between a curve and a chord.

오차벡터 $\mathbf{h}(t)$의 최대크기를 결정하기 위하여 일단 함수식 $F(t)$를 식 (2)와 같이 정의하자.

$$F(t) = |\mathbf{h}(t)|^2 = \mathbf{h}(t)\cdot\mathbf{h}(t) \qquad (2)$$

식 (1)과 식 (2)를 각각 미분하면 다음과 같다.

$$\dot{\mathbf{h}}(t) = \frac{d\mathbf{h}(t)}{dt} = \dot{\mathbf{r}}(t) - \{\dot{\mathbf{r}}(t)\cdot\mathbf{u}\}\mathbf{u} \qquad (3)$$

$$\frac{dF(t)}{dt} = 2\mathbf{h}(t)\cdot\dot{\mathbf{h}}(t) \qquad (4)$$

식 (3)을 식 (4)에 대입하면 다음의 결과를 얻게 된다.

$$\begin{aligned}\frac{dF(t)}{dt} &= 2\{\mathbf{h}(t)\cdot\dot{\mathbf{r}}(t) - [\dot{\mathbf{r}}(t)\cdot\mathbf{u}][\mathbf{u}\cdot\mathbf{h}(t)]\}\\ &= 2\mathbf{h}(t)\cdot\dot{\mathbf{r}}(t) \qquad (5)\end{aligned}$$

구간곡선의 최대오차는 식 (5)의 값이 0을 만족하는 곡선상의 점들 가운데 존재하므로 식 (6)을 곡선 $\mathbf{r}(t)$에 대한 최대오차조건이라고 부르기로 한다.

$$2\mathbf{h}(t)\cdot\dot{\mathbf{r}}(t) = 0 \quad (t_i \le t \le t_{i+1}) \qquad (6)$$

$t = t^*$에서 식 (6)의 최대오차조건이 만족되는 경우 구간곡선에서 오차벡터의 크기 $|\mathbf{h}(t^*)|$는 식 (1)로부터 다음과 같이 얻어진다.

$$|\mathbf{h}(t^*)| = \sqrt{|\mathbf{r}(t^*) - \mathbf{P}_i|^2 - \{[\mathbf{r}(t^*) - \mathbf{P}_i]\cdot\mathbf{u}\}^2} \qquad (7)$$

만약 주어진 구간 내에서 최대오차조건을 만족하는 점이 두개 이상 존재하는 경우에는 이들 오차의 크기들을 서로 비교하여 가장 큰 값을 구간곡선의 최대오차로 정한다.

3. 최대오차의 결정

3.1 평면곡선에 대한 최대오차의 결정

만약 $\mathbf{r}(t)$가 평면곡선인 경우 오차벡터 $\mathbf{h}(t)$는 항상 동일한 방향을 향하게 되므로 다음과 같이 표현될 수 있다.

$$\mathbf{h}(t) = d(t)\mathbf{H} \qquad (8)$$

여기서 $d(t) = |\mathbf{h}(t)|$

$\mathbf{H} = \mathbf{h}(t)/|\mathbf{h}(t)| = \mathbf{N}\times\mathbf{u}$

$\mathbf{N}$: 곡선이 존재하는 평면의 단위법선벡터

식 (8)을 식 (6)의 최대오차조건에 대입하면 다음을 얻게 된다.

$$d(t)\mathbf{H}\cdot\dot{\mathbf{r}}(t)=0 \quad (t_i \le t \le t_{i+1}) \tag{9}$$

$t = t_i$와 $t = t_{i+1}$에서는 오차벡터의 크기 $d(t)=0$이므로 최대오차는 존재하지 않는다. 따라서 식 (6)의 최대오차조건은 식 (10)과 같이 바꿀 수 있다.

$$\mathbf{H}\cdot\dot{\mathbf{r}}(t)=0 \quad (t_i \le t \le t_{i+1}) \tag{10}$$

즉 Fig. 2와 같이 최대오차를 만족하는 점에서는 단위오차벡터 **H**와 곡선의 접선벡터 $\dot{\mathbf{r}}(t)$가 수직으로 만난다.

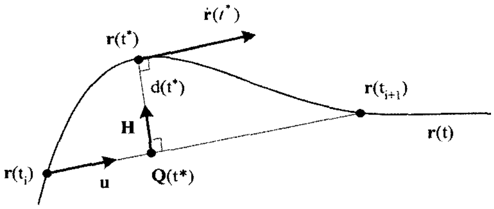


Fig. 2. A point of the maximum chordal deviation.

단위벡터 **H**는 t의 함수가 아니므로 곡선 $\mathbf{r}(t)$가 n차 다항식인 경우 식 (10)은 $n-1$차 다항식이 된다.

4차 이하의 다항방정식은 대수적인 방법으로 해를 구할 수 있으므로[4], 곡선 $\mathbf{r}(t)$가 5차 이하인 경우에는 주어진 구간에서 곡선에 대한 최대오차를 반복없이 한번에 구할 수 있다.

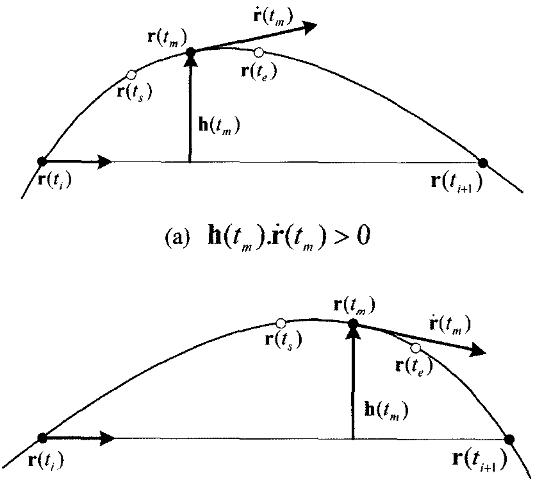
3.2 공간 곡선에 대한 최대오차의 결정

$\mathbf{r}(t)$가 공간곡선인 경우에는 평면곡선에서와 달리 식 (1)의 오차벡터 $\mathbf{h}(t)$는 식 (8)과 같이 표현될 수 없으므로 대수적으로 식 (6)의 최대오차조건을 만족하는 해를 구하는 것은 불가능해진다. 이 경우 구간 $[t_i, t_{i+1}]$에 존재하는 공간곡선 $\mathbf{r}(t)$에 대한 최대오차는 Fig. 3의 알고리즘을 따르는 이분탐색법을 사용하여 쉽게 구할 수 있다.

1. Set a search interval $[t_s, t_e] = [t_i, t_{i+1}]$
2. Find $t_m = (t_s + t_e)/2$
3. WHILE ($\lvert \mathbf{h}(t_m)\cdot\dot{\mathbf{r}}(t_m) \rvert > \varepsilon$) {
4. IF $\mathbf{h}(t_m).\dot{\mathbf{r}}(t_m) > 0$ THEN $t_s = t_m$
5. ELSE $t_e = t_m$
6. Find $t_m = (t_s + t_e)/2$
7. }
8. Find $\lvert \mathbf{h}(t_m) \rvert$ from Eq(7)

Fig. 3. An algorithm for finding the maximal deviation on a spatial curve segment.

Fig. 4(a)에서 보는 바와 같이 탐색구간의 중점에 대한 공간곡선 상의 점 $\mathbf{r}(t_m)$에서 곡선의 접선벡터 $\dot{\mathbf{r}}(t_m)$과 오차벡터 $\mathbf{h}(t_m)$의 사잇각이 90도보다 작은 예각을 이루는 경우, 즉 $\mathbf{h}(t_m)\cdot\dot{\mathbf{r}}(t_m)>0$인 경우에는 공간곡선 $\mathbf{r}(t)$의 최대오차는 구간 (t_m, t_e) 사이에 존재한다. 반면에 Fig. 4(b)와 같이 공간곡선 상의 점 $\mathbf{r}(t_m)$에서 곡선의 접선벡터 $\dot{\mathbf{r}}(t_m)$과 오차벡터 $\mathbf{h}(t_m)$의 사잇각이 90도보다 큰 둔각인 경우, 즉 $\mathbf{h}(t_m)\cdot\dot{\mathbf{r}}(t_m)<0$인 경우에는 최대오차는 구간 (t_s, t_m) 사이에 존재한다. 만약 곡선의 접선벡터 $\dot{\mathbf{r}}(t_m)$과 오차벡터 $\mathbf{h}(t_m)$의 사잇각이 90도에 충분히 가까워지는 경우, 즉 $\lvert\mathbf{h}(t_m)\cdot\dot{\mathbf{r}}(t_m)\rvert<\varepsilon$를 만족하게 되면 식 (7)을 이용하여 최대오차의 크기 $\lvert\mathbf{h}(t_m)\rvert$를 얻게 된다.



(a) $\mathbf{h}(t_m).\dot{\mathbf{r}}(t_m) > 0$

(b) $\mathbf{h}(t_m).\dot{\mathbf{r}}(t_m) < 0$

Fig. 4. Determination of a new search interval

4. 순차적 방법에 의한 점열의 생성

점열화 과정에서 허용오차는 최대한계값으로 주어지는 것이 일반적이다. 그러나 이렇게 허용오차를 정하는 경우 불필요하게 많은 점들이 생성될 수 있을 뿐만 아니라 얻어진 점들간의 간격 또한 균등하지 않게 될 가능성도 있다. 본 연구에서는 주어진 허용오차 범위 내에서 보다 적은 수의 점들로 이루어진 점열을 생성하기 위하여 최대허용오차값 ε_{max}와 최소허용오차값 ε_{min}을 갖는 허용오차의 범위 $[\varepsilon_{min}, \varepsilon_{max}]$를 정하기로 한다. 여기서 최대허용오차 ε_{max}은 지정된 허용오차에 해당하고, 최소허용오차의 크기는 $\varepsilon_{min} = \alpha\varepsilon_{max}$로 정한다. 최소허용오차를 결정하기 위한 α 값은 0과 1 사이의 값을 가지며 값이 0인 경우 최대허용오차만 주

어진 경우와 일치한다. 또한 α 값이 1에 가까울수록 점들 사이의 간격은 넓어지며 보다 적은 수의 점들이 얻어진다.

공간곡선 $\mathbf{r}(t)$로부터 점열을 순차적으로 얻는 과정을 설명해 보자. 곡선 상의 한점에서 곡률은 곡선의 길이에 대한 접선각도의 순간변화율로 정의된다[5]. 만약 곡선 상에 이웃한 두 점 $\mathbf{P}_i$, $\mathbf{P}_{i+1}$이 매우 가깝게 존재하는 경우, Fig. 5에 도시된 바와 같이 곡선 상의 이웃한 두 점 $\mathbf{P}_i$, $\mathbf{P}_{i+1}$와 그 점에서의 접선 $\dot{\mathbf{r}}(t_i), \dot{\mathbf{r}}(t_{i+1})$ 를 이용하여 곡률 κ는 식 (11)와 같이 추정할 수 있다[6].

$$\kappa = \Delta\theta/\Delta s \tag{11}$$

여기서 Δs: 이웃한 두 점 $\mathbf{P}_i$, $\mathbf{P}_{i+1}$간의 거리
$\Delta\theta$: 두 접선 $\dot{\mathbf{r}}(t_i), \dot{\mathbf{r}}(t_{i+1})$ 의 사잇각

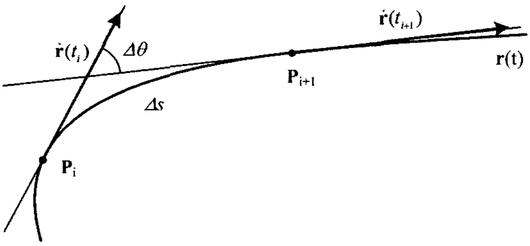


Fig. 5. Calculation of the curvature on a curve

Fig. 6은 현재의 점 $\mathbf{P}_i = \mathbf{r}(t_i)$ 에서 접선방향으로 step size d만큼 떨어져 있는 점 $\mathbf{G}$와 그 점에서 가장 가까운 곡선 상의 점 $\mathbf{P}^G = \mathbf{r}(t^G)$ 구하는 방법을 보여주고 있다. 먼저 점 $\mathbf{P}_i$에서 곡선에 접하면서 곡률반지름 $\rho = 1/\kappa$를 갖는 원호를 정한다. 또한 원호의 양

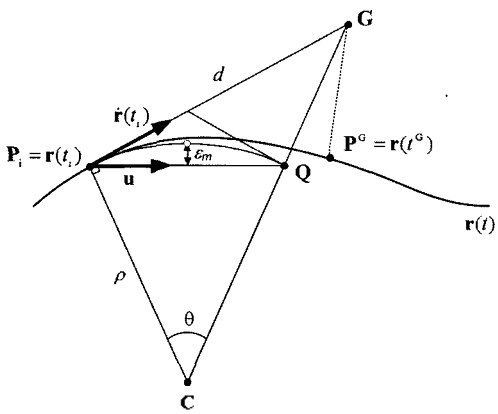


Fig. 6. The guess point **G** for a next point.

끝점 $\mathbf{P}_i$와 $\mathbf{Q}$를 연결한 현과 원호 사이의 오차가 평균 허용오차 $\varepsilon_m = (\varepsilon_{max} + \varepsilon_{min})/2$이 되도록 점 $\mathbf{Q}$를 정한다. 추정점 $\mathbf{G}$는 원호의 중심점 $\mathbf{C}$와 점 $\mathbf{Q}$를 연결한 직선과 점 $\mathbf{P}_i$에서의 접선이 만나는 점으로 결정된다.

Fig. 6으로부터 다음의 두 관계식을 얻을 수 있다.

$$d = \rho\tan\theta \tag{12}$$

$$\varepsilon_m = \rho(1 - \cos(\theta/2)) \tag{13}$$

여기서 ρ: 점 $\mathbf{P}_i$에서 곡률반지름 $1/\kappa$
$\varepsilon_m = (\varepsilon_{min} + \varepsilon_{max})/2$

식 (12)와 식 (13)에서 θ를 소거하면 step size d를 식 (14)와 같이 얻게 된다.

$$d = \frac{2(1-\kappa\varepsilon_m)\sqrt{2\kappa\varepsilon_m - \kappa^2\varepsilon_m^2}}{\kappa(1-4\kappa\varepsilon_m + 2\kappa^2\varepsilon_m^2)} \tag{14}$$

점 $\mathbf{G}$에 가장 가까운 곡선상의 추정점 $\mathbf{P}^G = \mathbf{r}(t^G)$은 식 (15)를 이용하여 근사적으로 얻을 수 있다[7].

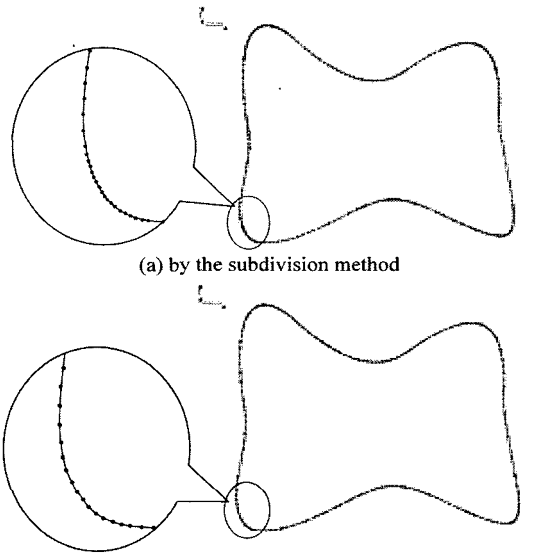
$$t^G = t_i + \Delta t \tag{15}$$

여기서 $\Delta t = d/|\dot{\mathbf{r}}(t_i)|$

얻어진 추정점 $\mathbf{P}^G$와 점 $\mathbf{P}_i$사이에서 구간곡선의 최대오차는 3.2절에서 이미 소개된 방법에 의하여 구할 수 있다. 만약 얻어진 최대오차의 크기가 허용오차의 범위 $[\varepsilon_{min}, \varepsilon_{max}]$내에 속한다면 다음 점 $\mathbf{P}_{i+1}$는 $\mathbf{P}^G$에 의하여 결정된다. 만약 추정점 $\mathbf{P}^G = \mathbf{r}(t^G)$의 최대오차의 크기 $|\mathbf{h}(t_m)|$가 허용오차의 범위를 벗어나는 경우 추정점 파라메터값 t^G를 근사적 비례관계식에 의하여 얻은 값 $t^G(\varepsilon_m/|\mathbf{h}(t_m)|)$로 대신하여 추정점 $\mathbf{P}^G$를 수정한다. 수정된 추정점으로부터 얻어진 최대오차가 허용오차 범위에 들어올 때까지 파라메터값을 반복하여 수정한다.

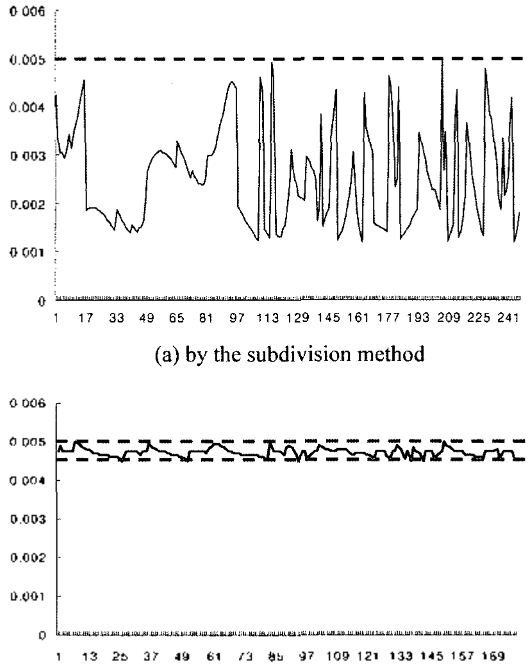
5. 분할법과 순차법의 비교

이제 분할법에 의하여 점열을 생성하는 방법과 순차법에 의하여 점열을 생성하는 방법을 비교해 보자. Fig. 7은 3차 Bezier 복합곡선으로부터 분할법과 순차법을 각각 사용하여 점열을 생성한 그림을 보여주고 있다. 여기서 허용오차의 범위는 [0.0045, 0.005]로 정하였다.


(a) by the subdivision method

(b) by the sequential method

Fig. 7. Comparison of two generated point sequences.



(a) by the subdivision method

(b) by the sequential method

Fig. 8. Chordal deviations of generated point sequences.

동일한 곡선에 대하여 Fig. 7(a)와 같이 분할적인 방법으로 생성된 점들의 개수는 242개이고, 순차법의 경우 생성된 점들의 수는 178개로 나타났다. 순차법이 분할법보다 약 26% 정도 적은 점들을 생성하는 것을 알 수 있다. 또한 그림에서 부분 확대된 부분을 보면 순차적 방법으로 생성된 점열에서 이웃한 점들간의 간격이 분할법의 경우보다 좀더 큰 것을 확인할 수 있다.

분할법과 순차적 방법으로 생성된 점열에 대하여 구간곡선에서의 최대오차를 도시하면 Fig. 8과 같다. 분할법의 경우 Fig. 8(a)에서 보는 바와 같이 구간곡선에서의 최대오차는 크게 차이를 보이고 있다. 특히 최대오차가 0의 값에 가까울수록 구간곡선의 길이는 매우 짧아지므로 생성된 점열은 많은 점들을 가지게 된다. 반면에 순차법의 경우 Fig. 8(b)에서 보는 바와 같이 지정된 허용오차의 범위 [0.0045, 0.005] 사이에 존재하게 되므로 상대적으로 적은 수의 점들로 이루어진 점열을 생성하게 된다.

그뿐만 아니라 분할법으로 점열을 생성할 경우에는 이진트리 데이터구조를 사용하여 점들을 생성하게 되므로 이진트리데이터구조가 필요없는 순차법에 비하여 상대적으로 많은 메모리를 필요로 하게 된다.

하지만 순차법은 분할법에 비하여 계산과정이 복잡하기 때문에 더 많은 계산시간을 필요로 한다. Fig. 7의 경우 분할법을 사용한 경우 점열을 구하는데 0.078초의 시간이 소요된 것에 비하여 순차법은 0.172초의 계산시간이 소요되었다. 그 외에 여러 곡선들에 대하여 두 방법을 비교조사해 본 결과 순차법이 분할법에 비하여 2~5배 정도 시간이 많이 소요되는 것으로 나타났다.

6. 결 론

지금까지 곡선으로부터 점열을 생성하는 방법으로는 허용오차를 만족할 때까지 곡선을 분할해 나가는 분할법이 이용되었다. 본 연구에서는 순차적인 방법을 사용하여 최대오차조건을 만족하는 점열의 생성방법을 제시하였다. 분할법으로 최대오차조건을 만족하는 점열생성의 방법에 비하여 순차법을 이용한 본 연구의 방법은 계산시간이 많이 소요되는 단점이 있지만 상대적으로 다음과 같은 장점이 있다.

첫째, 평면곡선은 물론 공간곡선에 대해서도 적용할 수 있다.

둘째, 순차적 방법은 분할법에 비하여 적은 수의 점을 가진 점열을 생성한다.

마지막으로, 순차적 방법은 분할법에 비하여 상대적으로 적은 메모리를 사용한다.

지금까지는 다항식곡선으로부터 점열을 생성하는

연구에 집중되었다. 차후에는 평면과 곡면의 교선 그리고 곡면과 곡면간의 교선으로부터 최대오차조건을 만족하는 점열을 생성하는 방법에 관한 연구가 필요하다.

감사의 글

이 논문은 2005년 울산대학교의 연구비에 의하여 연구되었습니다.

참고문헌

1. Anglada, M., Garcia, N. and Crosa, P., "Directional Adaptive Surface Triangulation", *Computer Aided Geometric Design*, Vol. 10, pp. 107-126, 1999.
2. Piegl, L. and Tiller, W., "Geometry-based Triangulation of Trimmed NURBS Surfaces", *Computer Aided Design*, Vol. 30, No. 1, pp. 11-18, 1998.
3. 주상윤, 추한, "평면곡선과 오프셋곡선의 점열화", 한국 CAD/CAM 학회 논문집, 제9권, 제2호, pp. 158-163, 2004.
4. de Boor, C., A Practical Guide to Splines, Springer-Verlag, New York, 1978.
5. Faux, I. D. and Pratt, M. J., Computational Geometric for Design and Manufacture, Ellis Horwood, Chichester, pp. 40-42, 1979.
6. 주상윤, 이상헌, "곡면간의 교선에서 Step Size 결정 및 접점탐지 방법", 한국 CAD/CAM 학회 논문집, 제3권, 제2호, pp. 121-126, 1998.
7. 주상윤, "곡면모델링에서 블렌딩곡면의 형성", KAIST박사학위논문, 1989.

주 상 윤

1977년 서울대학교 산업공학과 학사
1979년 한국과학원 산업공학과 석사
1989년 한국과학원 산업공학과 박사
1979년~현재 울산대학교 산업공학과 교수
관심분야: 곡면모델링, CAD/CAM, NC 가공, 제조시스템자동화